6.3 Change Control

6.3.1 Overview

Change control addresses the managing of the application environments such that configuration and development requests are identified and coordinated prior to movement. This includes movement within a Development environment, movement to a Quality Assurance environment, movement to the production environment, and building the Training environment. By the very nature of “change”, objects within the SCEIS solution will be configured during the activities within the Realization period. These configuration tasks include the defining of standard processes and information that require little to no adjustment subsequent to a Go Live event. These items include defining what time zone the organization operates under, how inventory pricing will be managed, how many characters the fund will be comprised of, etc. These items should not change during the life of the solution unless impacted by upgrades. Changes to this baseline setup require significant analysis and potential conversion.

6.3.2 Impact on SCEIS Projects

For implementation of the SCEIS, configuration activities will occur during each of the projects. The configuration activities will be generally limited to identifying the specific requirements and structures necessary for the individual agencies in that project to implement the SCEIS solution. The configuration activities do however require coordination to ensure the proper implementation. The complexity will occur in the configuration and change control subsequent to the initial Go Live of central government agencies (Project 4). Upon implementation of Project 4, the project team will continue their efforts in implementing the next agency group. At that point, application environments will already exist in:

- Development
- Quality Assurance and Testing
- Training
- Production

These application environments will support the agencies that are “live” within the SCEIS solution. The complexity occurs now that additional agencies will be configuring “their”

requirements within the SCEIS solution and movement of these changes will not and should not be transported into Production until the wave is “live”.

To add another layer of complexity to the change control process, the projects as identified in the implementation rollout schedule include the Business Blueprint for the Budget Preparation solution and the Human Resource/Payroll solution. These solutions will begin their blueprint and implementation activities prior to all state agencies being implemented with the finance and procurement SCEIS solution. Additional efforts are required to coordinate the following: the impacts and changes made within those component areas, the implementation of the applicable agency finance and procurement project, and the existing production environment.

6.3.3 Approach

The mechanism to support these types of processes is generally referred to as an “N+1” application environment. This refers to more than one development, testing and training environments. The separate environments exist to support each of their unique areas and will require a significant amount of testing to ensure changes customized in one application area for development purposes do not disrupt or ‘break’ functional areas in another application area and more importantly do not cause a problem within the production environment.

One approach for managing the “N+1” environments is to utilize the (SCEIS) production system as the original input for creation of the new development environment. Changes to the production system, created in the old development environment, can be manually established in the new development environment or potentially managed through the Solution Manager functionality. Depending on the training approach for educating users in agencies that have already gone live, the training environment will be required for the new development system and may or may not be required for the previous development system.

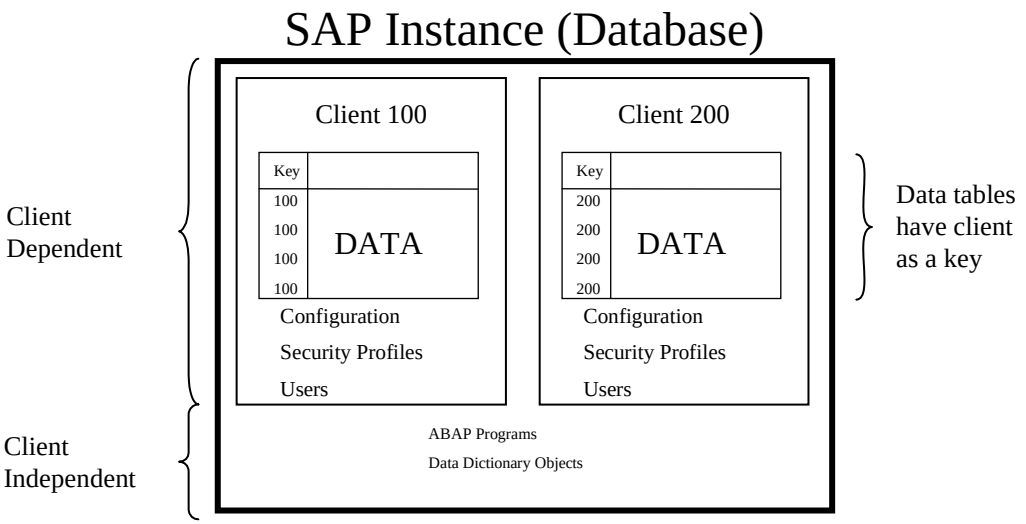
6.3.4 System Changes

Client Dependent and Client Independent

Changes to the application environment can be classified in one of two categories: Client Independent and Client Dependent. As the names indicate, Client Dependent changes affect only one client in an instance. These changes are typically to specific configuration settings, but can also include changes to authorization or security profiles as well as specific queries created within a client. Client Independent Changes affect all the clients within an instance. These changes generally consist of development activities to either modify a core component of the system or to

develop an enhancement to existing functionality. A key consideration in determining whether a "change" is client dependent or client independent is identified by whether the change has a direct impact to the data tables within the instance and whether those data tables include the data element "client" as a key to the table. A representation of dependent versus independent information is contained in the following exhibit.

Exhibit 6.3.4-1 Client Dependent v. Client Independent



Change Types

Changes can occur in a variety of ways within the confines of the implementation. Each change should be carefully planned and tested prior to movement into a production environment. Changes, as represented above, can be dependent or independent of the instance client(s). Therefore, a mechanism for change requests should be developed and followed during and after implementation.

Exhibit 6.3.4-2 Change Type Dependency

Change Type	Dependency
Custom Development/Enhancement	Client Independent
Modifications	Client Independent
Configuration/Customization	Client Dependent
Authorizations	Client Dependent
ABAP Queries	Client Dependent

Change Type	Dependency
SAP Script	Client Dependent
SAP Form	Client Dependent
Corrections - repairs to SAP delivered code (OSS Notes)	Client Independent

When changes are made within an SAP instance, the changes are generally transported (a migration process) from one instance to another or from one client to another. The functional process within the SCEIS solution involved in moving the changes is referred to as Transport Management. The use of Solution Manager within the SCEIS solution will assist the transport process by being able to identify the target systems that require the changes and automate some of the functions previously handled on a manual basis. Each object being transported requires a target instance and client. Client Independent objects will change the entire instance (all the clients), regardless of the client specified in the transport. Client dependent changes must be applied to each client in the target (recipient) system. Additional information on transport management can be found in the Technical section of this Business Blueprint.

6.3.5 Change Control Strategy

The system landscape strategy defined in the Technical section of this Business Blueprint identifies a proposed set of instances and clients to be maintained for the initial implementation efforts. These systems and clients will support the central government project (Project 4) from a development, testing, training and production basis. During the Realization efforts, the landscape will continue to be enhanced to accommodate new solutions and agencies in existing solutions. These include the additional solutions of Budget Preparation and Human Resources/Payroll and include the additional agency finance and procurement projects.

The change control strategy for the initial implementation activities will consist of some basic restrictions on the ability to initiate a transport request (ability to make configuration changes) and the ability to release a transport request (migrating the configuration changes from one instance to another or from one client to another). The project team will be responsible for coordinating these activities and help ensure the following: 1) configuration changes are appropriately made; and 2) changes that require predecessor activities to occur do not occur in an incorrect order.

When identifying the release of transport requests, additional considerations must be made. Transports made by the individuals conducting the configuration activities must be coordinated in the order that is appropriate for the tasks. The focus revolves around a transport that has been created to adjust a specific piece of configuration that is supportive of a functional area but not resident within the functional area. The responsible team may complete a transport with the accurate configuration settings when subsequently another transport overlays the correct settings. The coordination is lacking due to the flexibility of individuals to create and release their own transports. The recommended approach to managing the change is to allow a central point of contact within each team (for instance, the Team Lead) to manage the release of transports created by his/her team members.

From a functional perspective, most developers and configuration team members prefer the flexibility to release their own changes; however this permits changes to be released before they are scrutinized. Developers make a change, record the change in a request, finish their work, and release the change request. Developers prefer this model so that a specific change effort can be recorded and released as a version. Because transports lock some objects (pertaining mostly to ABAP code or other repository objects), releasing the lock immediately allows developers to start another version right away. If the transports are not requested to be imported into the recipient system in the order they were released, then it is possible that objects in transports can be overwritten with older versions. In addition, if any preliminary analysis is to be performed on the contents of the change request, it is best to do this before the change is released. Once the change request is released, the contents of the change request can no longer be modified. Allowing only the Team Leads to release the change requests force all changes to be evaluated before being released into the transport pipeline.

Exhibit 6.3.5-1 Pros and Cons to Team Lead Transport Release Approach

Pros	Cons
Guarantees only the latest development/configuration will be moved	More work for the Team Leads
Do not unknowingly overwrite newer development/ configuration in the system	Requires timely response from Team Leads to evaluate and release change requests
Less management from Development Teams	Requires the developer/configurator to wait until the transport is released before beginning additional work on the same objects in a change request (applicable to mostly repository objects).

Pros	Cons
Greater control over what is being released	
Makes the Team Lead more aware of the changes being released	
Forces the Team Lead to evaluate the change and approve it before releasing it	
Ongoing releases of the same program or configuration may give Team Lead an indication of an area to more closely evaluate for potential issues	
Closer analysis of changes before releasing can cut down potential number of transports	

The change control strategy will be developed during the Realization phase of the initial implementation. The change control strategy will need to address changes within the "N" level of the implementation. That is, how transport requests will be managed during the initial configuration activities. The strategy will also need to address each environment that will be managed for updates and synchronization upon successful implementation of "N". These types of environments include:

- N+1 Development system(s): These systems include the individual components of the SCEIS solution that are preparing for a Go Live with their respective project. These agencies include the next set of users of the SCEIS solution. Changes to configuration, to address the specific agency's needs, will be required. These changes will be made in the N+1 environment and must be coordinated amongst the project team members.
- Movement to N+1 Quality Assurance System: Changes to configuration within the N+1 development environment will be moved to the N+1 Quality Assurance environment for testing purposes. The changes need to be managed for the proper migration ordering and tested thoroughly within the environment. Changes that fail will then require additional migration work to adopt the appropriate configuration changes. Procedures will need to be in place to address the approval and support of movement from N+1 Development to Quality Assurance.

- Movement to Production: Movement to production requires an additional level of scrutiny. The changes being made within the N+1 Development environment and migrated to N+1 Quality Assurance will then require movement to production. The timing and authorization of the migration is critical. Movements to production should be carefully managed and approved following significant testing and retesting efforts. The N+1 activities are ultimately updating a production environment that will have been live with a number of users. The cutover and Go Live of new agencies will require their configuration changes to be moved at the appropriate time and in the correct order.
- Creation, Copy, and Cutoff for Training: Training instances will reflect copies of the Production or Quality Assurance environments as appropriate within the timing of the project. The Training environment will require a firm cutoff from updates in order to prepare for the requisite training classes. The usage of one or multiple training environments will be addressed as part of the strategy definition.
- Impact on Existing Users: The movement of changes to the production environment, in support of new implementations, may have an impact on existing users. To the extent the changes impact existing users and processes, communication of the changes and plans for re-training the existing users need to be considered. The change control strategy will reference the communications activities that will be put in place to notify and educate the existing users on changes to their production system.

For each of these key areas the change control strategy will need to provide answers and a process for the following questions:

- How will the committee or approval body be organized within the project team to approve movement of changes?
- How does change control impact or be impacted by the cutover from an N or N+1 environment to production?
- What clients are required within the SCEIS solution?
- How are changes received in the respective clients (client copy, manual entry, transport, etc.)?
- Who is authorized to create a transport?
- Who is authorized to release a transport?

- What signoff tools are required for transport requests?
- Who and how will transport requests be validated and tested?
- How does the transport request interact with requests being made from other solutions (BPS and HR/Payroll)?
- Who is responsible for coordinating these requests and their final migration?
- How will requests (development and corrective actions) be managed in the production environment?
- What is the impact of the changes to existing users in the production environment?
- How will communication and re-training activities be addressed when changes impact existing users?